ГЛАВА XB. Качество классификации на несколько классов и сравнение множеств

Ошибки для того и существуют, чтобы их делали.

С. Тартаковер

Когда смерил, так и поверил.

До настоящего момента оценка качества моделей рассматривалась в задачах регрессии и бинарной классификации. Рассмотрим теперь задачу с l классами, $l>2$, объекты выборки, на которой оцениваем качество, по-прежнему обозначаем $x_1,\ldots,x_m$. Классы могут пересекаться, тогда задача называется **задачей с l пересекающимися классами (с «мульти-метками», «Multi-label»)** и классификация одного объекта x_i описывается l-мерным бинарным вектором:

$$y(x_i)=(y_{i1},\ldots,y_{il})\in\{0,1\}^l, \qquad \text{(XB.1)}$$

в котором $y_{ij}=1$ тогда и только тогда, когда объект принадлежит j-му классу, а алгоритм выдаёт оценки принадлежности к каждому классу

$$b(x_i)=(b_{i1},\ldots,b_{il})\in\mathbb{R}^l \qquad \text{(XB.2)}$$

(иногда дополнительно требуют, чтобы все оценки лежали на отрезке $[0,1]$). Если классы не пересекаются, т.е. рассматриваем **задачу классификации с непересекающимися классами (multi class)**, тогда каждый объект принадлежит ровно одному классу:

$$y(x_i)\in\{1,2,\ldots,l\}$$

(логично классы перенумеровать и использовать номера в качестве меток), впрочем можно использовать и бинарные векторы (XB.1), иногда они полезны для записи формул, в каждом векторе классификации тогда будет ровно одна единичная компонента. В этом случае также алгоритм может вычислять оценки (XB.2), только здесь логично требовать условия нормировки:

$$b_{i1}+\ldots+b_{il}=1,\ b_{i1}\geq 0,\ldots,b_{il}\geq 0. \qquad \text{(XB.3)}$$

Чёткую классификацию из оценок алгоритм получает с помощью решающего правила, в задаче с непересекающимися классами логично относить объект к классу с наибольшей оценкой:

$$a(x_i) = \arg\max_t (b_{it}),$$

в задаче с пересекающимися классами простейший способ[1] отнесения к классам – пороговая бинаризация:

$$a(x_i) = I[b(x_i) \geq \theta] \equiv (I[b_{i1} \geq \theta], \ldots, I[b_{il} \geq \theta]).$$

Для оценки качества в задаче с непересекающимися классами часто строится **матрица несоответствий / ошибок (Confusion Matrix)**: матрица $M = \| m_{ij} \|_{l \times l}$ размера $l \times l$, в которой ij-й элемент равен числу объектов i-го класса, которым алгоритм присвоил метку j-го класса:

$$m_{ij} = \sum_{t=1}^{m} I[y_t = i] \cdot I[a_t = j],$$

заметим, что сумма всех элементов матрицы совпадает с числом объектов:

$$\sum_{i=1}^{l} \sum_{j=1}^{l} m_{ij} = m.$$

На рис. XB.1 показана матрица несоответствий для выборки из 10 объектов. Многие введённые ранее показатели качества естественным образом обобщаются на многоклассовый случай. Например, **доля верных ответов или точность[2] (accuracy)** по-прежнему равна

$$\text{Accuracy} = \frac{1}{m} \sum_{i=1}^{m} I[a_i = y_i] = \frac{1}{m} \sum_{i=1}^{m} m_{ii}.$$

Понятия TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative) теперь зависят от того, какой класс считать положительным, поэтому число истинно положительных для t-го класса:

В многоклассовой задаче многие понятия вводятся для каждого класса.

$$\text{TP}_t = m_{tt},$$

[1] Далее рассмотрим более сложные решающие правила.

[2] Термин точность также используется для перевода показателя качества Precision, поэтому возникает путаница. В последнее время часто «Accuracy» переводят как «аккуратность».

аналогично

$$\mathrm{FN}_t = m_{t1} + \ldots + m_{t,t-1} + m_{t,t+1} + \ldots + m_{t,l} = \sum_{j=1}^{l} m_{tj} - m_{tt},$$

$$\mathrm{FP}_t = m_{1t} + \ldots + m_{t-1,t} + m_{t+1,t} + \ldots + m_{l,t} = \sum_{i=1}^{l} m_{it} - m_{tt},$$

$$\mathrm{TN}_t = \sum_{\substack{1 \le i,\, j \le l \\ i \ne t,\, j \ne t}} m_{ij}.$$

Отсюда выводятся понятия полнота и точность для t-го класса:

$$R_t = \frac{\mathrm{TP}_t}{\mathrm{TP}_t + \mathrm{FN}_t} = \frac{m_{tt}}{m_{t1} + \ldots + m_{tl}},$$

$$P_t = \frac{\mathrm{TP}_t}{\mathrm{TP}_t + \mathrm{FP}_t} = \frac{m_{tt}}{m_{1t} + \ldots + m_{lt}},$$

среднее гармоническое которых будет **$\mathbf{F_1}$-мерой для t-го класса[1].**

	y	a
0	1	1
1	1	1
2	1	2
3	2	1
4	2	3
5	3	2
6	3	3
7	3	3
8	1	2
9	2	2

y \ a	1	2	3
1	2	2	0
2	1	1	1
3	0	1	2

	0	1	2
0	1.2	1.6	1.2
1	0.9	1.2	0.9
2	0.9	1.2	0.9

Рис. XB.1. Ответы и истинные метки (слева) и соответствующая им матрица несоответствий (по центру) и матрица несоответствий случайных ответов (справа).

[1] Перечисленные показатели выводятся с помощью функции sklearn.metrics.classification_report в sklearn.

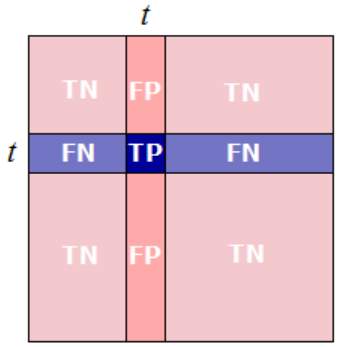

Рис. XB.2. Иллюстрация показателей TP, FN, FP и FN для t-го класса.

На основе матрицы несоответствий также вводится показатель качества **Weighted kappa**:

$$\kappa = 1 - \frac{\sum_{i=1}^{l}\sum_{j=1}^{l} w_{ij} m_{ij}}{\sum_{i=1}^{l}\sum_{j=1}^{l} w_{ij} s_{ij}} \in [-1, +1], \qquad \text{(XB.4)}$$

где $w_{ij} \in \mathbb{R}^+$ – штраф за отнесение объекта i-го класса j-му классу, а $S = \| s_{ij} \|_{l \times l}$ – матрица несоответствий случайных ответов:

$$s_{ij} = \frac{\sum_t m_{it} \sum_t m_{tj}}{m}.$$

Матрица S обладает следующими свойствами:

- состоит из неотрицательных элементов (как матрица M, правда, не обязательно целых),
- сумма всех элементов равна числу объектов m (как матрица M),
- сумма всех строк равна сумме всех строк матрицы M (т.е. матрица M соответствует такому же распределению меток в ответах),
- сумма всех столбцов равна сумме всех столбцов матрицы M (т.е. матрица M соответствует такому же распределению истинных меток).

Пример матрицы случайных ответов S показан на рис. XB.1 (суммы всех строк равны вектору $(3,4,3)$, а столбцов – вектору $(4,3,3)^{\mathrm{T}}$). Матрица S используется для нормировки в (XB.4), чтобы случайные ответы (если наш алгоритм

Нормировка на качество случайного ответа – стандартный приём.

случайно генерирует метки при таких же распределениях меток-ответов) соответствовали около нулевому качеству.

Штрафы $w_{ij} \in \mathbb{R}^+$ обычно выбираются исходя из постановки задачи, например в показателе **Quadratic Weighted Kappa** квадратичными:

$$w_{ij} = (i-j)^2,$$

что логично, когда классы линейно упорядочены и этому порядку соответствуют их номера (например, в задаче оценки релевантности двух текстов: 1 – «плохо», 2 – «не очень», 3 – «хорошо», 4 – «отлично»). Тогда сильнее штрафуется отнесение объекта класса 4 ко 2-му классу, чем к 3-му, см. рис. XB.2.

	y	1.0	0.83	0.83	0.33	0.8	0.0	-1.0
0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	1	2
2	0	0	1	0	2	0	2	2
3	1	1	1	1	1	0	0	1
4	1	1	1	1	1	0	1	1
5	1	1	0	2	1	0	2	1
6	2	2	2	2	2	2	0	0
7	2	2	2	2	2	2	1	0
8	2	2	2	1	0	2	2	0

Рис. XB.2. Значения Quadratic Weighted Kappa при разных ответах.

Вернёмся к проблеме оценивания качества в задаче классификации с несколькими классами. Обратим внимание, что классификация выборки из m объектов задаётся бинарной матрицей – **матрицей классификаций**:

$$Y = \| y_{ij} \|_{m \times l} \in \{0,1\}^{m \times l},$$

$y_{ij} = 1$ тогда и только тогда когда объект x_i принадлежит j-му классу, а ответы алгоритмов на выборке – **матрицей ответов**:

$$A = \| a_{ij} \|_{m \times l} \in \mathbb{R}^{m \times l},$$

a_{ij} – оценка принадлежности объекта x_i j-му классу или бинарный ответ[1] (которую получил алгоритм).

Когда мы оценивали качество в задаче бинарной классификации, то фактически научились сравнивать похожесть двух бинарных векторов, например вычислять F_1-меру по вектору $y \in \{0,1\}^m$ меток объектов выборки и полученному вектору $a \in \{0,1\}^m$ оценок принадлежности к классу 1, а также похожесть бинарного вектора на вещественный, например вычислять $\mathrm{ROC}_{\mathrm{AUC}}$ по $y \in \{0,1\}^m$ и $a \in \mathbb{R}^m$. Теперь надо оценить сходство матриц Y и A. Для этого есть следующие стандартные способы, выбрать показатель качества бинарной классификации $F(y,a)$ и сделать

Матрицы можно сравнить как векторы, по строкам, по столбцам. Также можно использовать веса.

- **микро-усреднение (Micro-averaging):**

  векторизовать матрицы Y и A, вычислить качество по таким векторам,

$$F_{\mathrm{micro}} = F((y_{11}, y_{12}, \ldots, y_{ml}), (a_{11}, a_{12}, \ldots, a_{ml})).$$

- **усреднение по объектам:**

  усреднить схожести строк матриц:

$$F_{\mathrm{samples}} = \frac{1}{m}\sum_{i=1}^{m} F((y_{i1}, y_{i2}, \ldots, y_{il}), (a_{i1}, a_{i2}, \ldots, a_{il})).$$

- **макро-усреднение (усреднение по классам, Macro-averaging)**

  усреднить[2] схожести столбцов матриц:

$$F_{\mathrm{macro}} = \frac{1}{l}\sum_{j=1}^{l} F((y_{1j}, y_{2j}, \ldots, y_{mj}), (a_{1j}, a_{2j}, \ldots, a_{mj})),$$

  т.е. посчитать качество для каждого класса в отдельности, а затем усреднить.

- **взвешенное макро-усреднение**

  применяют, когда каждому классу приписан некоторый вес:

[1] Здесь и ниже не будем отдельно рассматривать матрицу оценок и матрицу ответов (т.к. подходы для определения по ним качества схожи).

[2] Иногда встречаются и другие способы усреднения, например, среднее геометрическое.

$$F_{\text{weighted}} = \frac{1}{P_1 + \ldots + P_l} \sum_{j=1}^{l} P_j F((y_{1j}, y_{2j}, \ldots, y_{mj}), (a_{1j}, a_{2j}, \ldots, a_{mj})),$$

чаще всего вес пропорционален числу / доле объектов с меткой соответствующего класса в обучающей выборке: $P_j = y_{1j} + y_{2j} + \ldots + y_{mj}$.

В табл. XB.1 показаны значения различных обобщений площади под ROC-кривой (ROC_{AUC}) на случай нескольких классов (численно они не совпадают).

Матрица классификаций

	class 1	class 2	class 3
0	1	1	0
1	0	1	1
2	0	1	0
3	1	0	0

Матрица ответов

	class 1	class 2	class 3
0	0.7	0.6	0.5
1	0.3	0.4	0.6
2	0.5	0.9	0.2
3	0.4	0.5	0.1

Разные варианты ROC_{AUC}

macro	weighted	micro	samples
0.806	0.75	0.833	0.875

ROC_{AUC} по классам, веса классов, по объектам

class 1	0.750
class 2	0.667
class 3	1.000

P_1	0.50
P_2	0.75
P_3	0.25

x_1	1.0
x_2	1.0
x_3	1.0
x_4	0.5

Табл. XB.1. Различные усреднения ROC_{AUC} на конкретном примере.

Кроме описанных общих подходов к сравнению матриц Y и A, также используют **долю ошибок / ошибку Хэмминга (Hamming Loss)** для сравнения бинарных матриц ($a_{ij} \in \{0,1\}$):

$$\text{HL} = \frac{1}{ml} \sum_{i=1}^{m} \sum_{j=1}^{l} I[y_{ij} \neq a_{ij}].$$

В задачах с непересекающимися классами с условиям нормировки (XB.3) в ответе иногда используется функционал **Mean Probability Rate**:

$$\text{MPR} = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{l} y_{ij} a_{ij},$$

а также макро-версия этого показателя качества:

$$\text{MAPR} = \frac{1}{l}\sum_{j=1}^{l}\frac{\sum_{i=1}^{m} y_{ij}a_{ij}}{\sum_{i=1}^{m} y_{ij}}.$$

Также для оценки сходств матриц привлекают функции ошибки, которые традиционно используются в регрессии:

$$\text{MSE} = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}(y_{ij} - a_{ij})^2,$$

$$\text{MAE} = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}|\, y_{ij} - a_{ij}\,|.$$

Логистическая функция ошибки довольно просто обобщается на случай многих классов. В задаче с пересекающимися классами чаще используют макро-усреднение logloss:

$$\text{logloss}_{\text{multi-label}} = -\frac{1}{l}\frac{1}{m}\sum_{j=1}^{l}\sum_{i=1}^{m}(y_{ij}\log a_{ij} + (1-y_{ij})\log(1-a_{ij})),$$

здесь всё понятно: для каждого класса независимо считается логистическая ошибка, такую функцию ошибки называют также **бинарной кросс-энтропией**. В задаче с непересекающимися классами в матрице ответов есть условие нормировки (XB.3) и использует обобщение логистической функции или **кросс-энтропия**:

$$\text{logloss} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij}\log a_{ij}\,.$$

Если считать, что у каждого объекта x_i выборки есть истинная метка $y_i \in \{1,2,\ldots,l\}$, то формула перепишется в виде

$$\text{logloss} = -\frac{1}{m}\sum_{i=1}^{m}\log a_{i,y_{ij}}\,.$$

Кросс-энтропия = усреднение минус логарифмов от оценок вероятностей принадлежности к истинным классам.

Сбалансированная точность «Balanced accuracy» по аналогии с бинарной классификацией вводится как макро-усреднение полноты[1]:

$$\text{BA} = \frac{1}{l}\sum_{j=1}^{l} R_j = \frac{1}{l}\sum_{j=1}^{l} \frac{\sum_{i=1}^{m} I[y_{ij}=1]\cdot I[a_{ij}=1]}{\sum_{i=1}^{m} I[y_{ij}=1]}.$$

[1] Иногда в литературе под сбалансированной точностью понимают усреднение по классам минимума точности и полноты или минимума чувствительности и специфичности.

Сравнение множеств

Предположим, что значения целевого признака – множества. В задаче классификации с пересекающимися классами это как раз так, поскольку каждому объекту соответствует множество меток. Но, вообще говоря, множества могут быть и более сложные, например бесконечные. Для определённости рассмотрим случай, когда множества отрезки:

$$\{(x_i,[y_{i1}, y_{i2}])\}_{i=1}^m$$

$y_{i1} \le y_{i2}$ для всех $i \in \{1,2,\ldots,m\}$. Примером задачи с такими целевыми значениями является определение диапазона цен на квартиры в районе по описанию характеристик квартир и района или определение диапазона значений долей химических элементов при выплавке стали. Чтобы оценить качество ответов на выборке

$$[a_{11}, a_{12}],[a_{21}, a_{22}],\ldots,[a_{m1}, a_{m2}]$$

необходимо научиться сравнивать два множества, т.е. придумать функцию близости $F(A,B)$ от множеств A и B. Логично потребовать, чтобы

$$0 = F(A,A) < F(A,B) \text{ при } A \cap B \neq \varnothing.$$

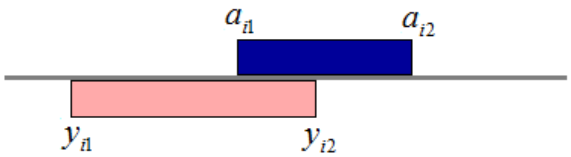

Рис. XB.3. Два интервала: целевой и ответ алгоритма.

Подходящих функций довольно много, они называются **коэффициентами сходства**[1] **(мерами / индексами сходства)**, вот некоторые из них, основанные на мощностях множеств:

коэффициент Жаккара (Jaccard) – $\dfrac{|A \cap B|}{|A \cup B|}$,

коэффициент Шимкевича-Симпсона (Szymkiewicz, Simpson) – $\dfrac{|A \cap B|}{\min(|A|,|B|)}$,

[1] Интересно, что традиционно они использовались в биологии, географии, социологии. Например, для оценивания сходства одной популяции до и после некоторого периода.

коэффициент Браун-Бланке (Braun-Blanquet) – $\frac{|A\cap B|}{\max(|A|,|B|)}$,

коэффициент Сёренсена (Sörensen) – $\frac{2|A\cap B|}{|A|+|B|}$,

коэффициент Кульчинского (Kulczinsky) – $\frac{|A\cap B|}{2}\frac{1}{1/|A|+1/|B|}$,

коэффициент Отиаи (Ochiai) – $\frac{|A\cap B|}{\sqrt{|A|\cdot|B|}}$,

Есть также несимметричные функционалы, т.н. «меры включения»:

$$\frac{|A\cap B|}{|A|},\qquad \frac{|A\cap B|}{|B|},$$

$$\frac{|A\cap B|}{2|A|-|A\cap B|},\qquad \frac{|A\cap B|}{2|B|-|A\cap B|}.$$

Для сравнения конечных множеств (также, например, строк) иногда используют **«редакторское расстояние»**: описывают операции, которые изменяют множества и расстоянием называют наименьшее число операций, с помощью которых одно множество можно превратить в другое.

Для примера рассмотрим проблему оценки качества кластеризатора на наборах данных с известными референсными кластеризациями. Для каждого набора данных $x_1,\ldots,x_m$ (мощность m может меняться от набора к набору) известно референсное разбиение на кластеры: $K_1\cup\ldots\cup K_t=\{1,2,\ldots,m\}$, $K_i\cap K_j=\varnothing$ при $i\neq j$. Есть ответ алгоритма в виде разбиения: $C_1\cup\ldots\cup C_s=\{1,2,\ldots,m\}$, $C_i\cap C_j=\varnothing$ при $i\neq j$, необходимо эти два разбиения сравнить. Определим набор операций:

- add(C,x) – добавление одного объекта x к кластеру C,
- create(x) – создание кластера с одним объектом x,
- del(C,x) – удаление одного объекта x из кластера C,
- erase(C) – удаление кластера с одним объектом (т.е. $C=\{x\}$).

Например, редакторское расстояние между

$$\tilde{C} \sim \{1,2,3\} \cup \{4,5\} \cup \{6\},$$

$$\tilde{K} \sim \{1\} \cup \{2,3\} \cup \{4,5,6\}$$

равно 4:

$$\{1,2,3\} \cup \{4,5\} \cup \{6\} \xrightarrow{\text{erase}}$$

$$\{1,2,3\} \cup \{4,5\} \xrightarrow{\text{del}}$$

$$\{2,3\} \cup \{4,5\} \xrightarrow{\text{create}}$$

$$\{1\} \cup \{2,3\} \cup \{4,5\} \xrightarrow{\text{add}}$$

$$\{1\} \cup \{2,3\} \cup \{4,5,6\}$$

Приложения и примеры

1. Рассмотрим различные **обобщения точности на многоклассовый случай**, макро-точность:

$$\mathrm{P}_{\text{macro-mean}} = \frac{1}{l}\sum_{j=1}^{l} \mathrm{P}_j \text{ , где } \mathrm{P}_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j},$$

и микро-точность:

$$\mathrm{P}_{\text{micro-mean}} = \frac{\sum_{j=1}^{l} \mathrm{TP}_j}{\sum_{j=1}^{l} \mathrm{TP}_j + \sum_{j=1}^{l} \mathrm{FP}_j}.$$

Естественно, разные обобщения численно могут не совпадать. Рассмотрим две задачи, у которых не различаются точности по классам, см. рис. XB.4, поэтому у них не различаются и макро-точности, равные

$$\mathrm{P}_{\text{macro-mean}} = \frac{1}{3}\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5}\right] \approx 0.344,$$

но микроточность в первой задаче равна

$$P_{\text{micro-mean}} = \frac{2+5+10}{4+15+50} \approx 0.246,$$

а во второй –

$$P_{\text{micro-mean}} = \frac{2+5+100}{4+15+500} \approx 0.206.$$

Задачи отличаются тем, что во второй увеличилось в 10 раз число объектов одного из классов, в итоге микро-точность сместилась к значению точности за этот класс. Такие свойства микроусреднения надо учитывать в реальных задачах (полезно также смотреть на дисперсию показателей качества по разным классам).

	TP	FP
class 1	2	2
class 2	5	10
class 3	10	40

	TP	FP
class 1	2	2
class 2	5	10
class 3	100	400

	P
class 1	0.50
class 2	0.33
class 3	0.20

Рис. XB.4. Значения TP и FP в двух задачах (слева и по центру) и точности по классам (справа) в этих задачах.

2. Часто приходится **уточнять, что имеется в виду при использовании термина**. Например, макро-F-мерой логично назвать макро-усреднение F-мер по классам:

$$F_{\text{macro-mean}} = \frac{1}{l}\sum_{j=1}^{l} F_j ,$$

а можно назвать – среднее гармоническое макро-точности и макро-полноты:

$$F = \frac{2}{\frac{1}{P_{\text{macro-mean}}} + \frac{1}{R_{\text{macro-mean}}}}.$$

3. Выше была описана **задача с метками, которые являются отрезками**. Опишем несколько стандартных способов её решения. Предположим, что показатель качества выбран, например это коэффициент Жаккара (Jaccard)

$$F(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

Отрезок задаётся двумя параметрами, поэтому естественно решать две задачи регрессии (с такими же обучающими объектами, но новыми вещественными метками). Причём в качестве целевых значений для этих задач можно выбрать любую пару из списка:

- начало отрезка (левый конец),
- окончание отрезка (правый конец),
- середина отрезка,
- ширина отрезка.

Решать задачи с новыми числовыми целевыми признаками можно стандартными моделями, но тогда не учитывается специфика выбранного показателя качества. Для его оптимизации подходит концепция решающих правил: пусть есть предварительный ответ $[\alpha, \beta]$, формируем параметрическую деформацию ответа в виде

$$C_\varepsilon([\alpha,\beta]) = \left[\frac{\alpha+\beta}{2} - \varepsilon\frac{\beta-\alpha}{2}, \frac{\alpha+\beta}{2} + \varepsilon\frac{\beta-\alpha}{2}\right].$$

Заметим, что $C_1([\alpha,\beta]) = [\alpha,\beta]$, а $C_0([\alpha,\beta])$ является множеством, состоящим из одной точки – середины отрезка $[\alpha,\beta]$. Гиперпараметр $\varepsilon \in \mathbb{R}^+$ можно настраивать прямым перебором, оптимизируя в явном виде заданный показатель качества (на отложенной выборке). Отметим также, что деформация может учитывать естественные ограничения на ответ, например, при прогнозе диапазона цен границы диапазона не могут быть отрицательными, что можно учесть при деформации:

Обучать алгоритм и настраивать решающее правило лучше на разных выборках.

$$C_\varepsilon([\alpha,\beta]) = \left[\max\left[\frac{\alpha+\beta}{2} - \varepsilon\frac{\beta-\alpha}{2}, 0\right], \frac{\alpha+\beta}{2} + \varepsilon\frac{\beta-\alpha}{2}\right].$$

Напомним схематично **концепцию решающего правила**:

- есть базовые алгоритмы (операторы),
- есть параметризованный способ перевода (деформации) их ответов в нужные (такой перевод и осуществляет т.н. «решающее правило»),

- производится прямая минимизация целевого функционала с помощью подбора параметров деформации.

4. В **задачах классификации с несколькими классами** часто бывает полезно применять концепцию решающих правил для получения чёткой классификации по матрице оценок (по сути, по матрице нечётких классификаций). Простейшая стратегия бинаризации матрицы оценок – сравнение каждого элемента b_{ij} (оценки принадлежности i-го объекта к j-му классу) с некоторым порогом $\theta \in \mathbb{R}$ (который подбирается оптимизируя целевой показатель качества). При использовании такого решающего правила

Задачу можно решать «по вертикали»: для каждого класса определять, какие объекты к нему отнести, а можно «по горизонтали»: для каждого объекта определять, к каким классам его отнести.

$$a_{ij} = \begin{cases} 1, & b_{ij} \geq \min(\theta, \max\{b_{ij}\}_{j=1}^{l}), \\ 0, & \text{иначе} \end{cases}$$

(здесь a_{ij} – бинарный ответ на вопрос о принадлежности i-го объекта к j-му классу) гарантируется, что каждый объект будет отнесён хотя бы к одному классу. В одном из соревнований[1] по машинному обучению качество существенно повышалось при обеспечении гарантии отнесения к каждому классу некоторой доли объектов.

5. В главе описана оценка качества кластеризаций при заданных референсных. Задача построения таких кластеризаций возникает, когда требуется выполнять разбиения на кластеры на разных наборах данных, при этом есть несколько примеров «хороших кластеризаций». В соревновании «Learning Social Circles in Networks[2]» необходимо было по эго-сети пользователя социальной сети (графу его друзей) определить, как он может объединить друзей в т.н. социальные круги (родственники, друзья по школе, вузу, лучшие друзья и т.п.). Причём для оценки решений использовалась функция ошибки на основе описанного редакторского расстояния. Если «затачиваться» на такую функцию, то получаются «осторожные решения», поскольку отнесение объекта в чужой кластер штрафуется дважды (одна операция – извлечь объект из кластера, вторая – положить в нужный), поэтому многие объекты алгоритм не относит ни в один кластер (такое было допустимо в данной задаче).

[1] https://www.kaggle.com/competitions/lshtc/

[2] https://www.kaggle.com/competitions/learning-social-circles

Задачи и вопросы

1. При описании показателя weighted kappa была введена матрица несоответствий случайных ответов S, а также описаны её свойства. Можно ли описать класс матриц, которые удовлетворяют таким условиям? Когда матрица S является однозначно определённой?

2. Докажите, что показатель каппа Коэна является частным случаем показателя weighted kappa.

3. В предложенном способе решения задачи с размеченными данными, в которых метки являются отрезками, перечислено 4 возможных целевых признака для двух задач регрессии. Какие из целевых признаков в каких ситуациях удобнее? Может ли итоговое качество сильно зависеть от выбора пары целевых признаков?

4. Как эффективно вычислять описанное редакторское расстояние над различными кластеризациями?

5. Как при бинаризации матрицы оценок гарантировать отнесение к каждому классу некоторой доли объектов? Выпишете соответствующее решающее правило. Как корректно подбирать доли для всех классов на практике?

6. В каких практических ситуациях лучше использовать макро-точность, а в каких микро-точность?

Классификация на несколько классов: итоги[1]

1. В задача с непересекающимися классами ошибки описываются матрицей несоответствий с элементами

$$m_{ij} = \sum_{t=1}^{m} I[y_t = i] \cdot I[a_t = j].$$

2. Если выбрать показатель качества для бинарной задачи классификации, то его естественные обобщения на многоклассовый случай:

- микро-усреднение,
- усреднение по объектам,
- макро-усреднение (часто используется по умолчанию),
- взвешенное макро-усреднение.

3. Взвешенная каппа (weighted kappa) является обобщением каппы Коэна и хорошо применима, когда есть матрица штрафов за отнесения объектов к неправильным классам.

4. Логистическая функция ошибки имеет два основных обобщения: бинарная кросс-энтропия

$$\text{BCE} = -\frac{1}{l}\frac{1}{m}\sum_{j=1}^{l}\sum_{i=1}^{m}(y_{ij}\log a_{ij} + (1-y_{ij})\log(1-a_{ij})),$$

и кросс-энтропия

$$\text{CE} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij}\log a_{ij}\ .$$

5. Для сравнения множеств есть много функций сходства, самая популярная – коэффициент Жаккара (Jaccard)

$$\frac{|A \cap B|}{|A \cup B|}.$$

[1] См. также хороший обзор Grandini M., Bagli E., Visani G. Metrics for multi-class classification: an overview //arXiv preprint arXiv:2008.05756. – 2020.

6. Очень полезна концепция решающего правила, в которой ответ меняется с помощью параметрического преобразование. Значение параметра подбирается с помощью оптимизации целевого функционала качества.

Спасибо за внимание к книге!
Замечания по содержанию, замеченные ошибки и неточности можно написать в телеграм-чате
https://t.me/Dyakonovsbook